CECS 447: Microprocessors and Controllers III

Project 1: Digital to Analog Converter

Jonic Mecija

ID: 014467048

February 27, 2019

Table of Contents

Table of Figures

Introduction

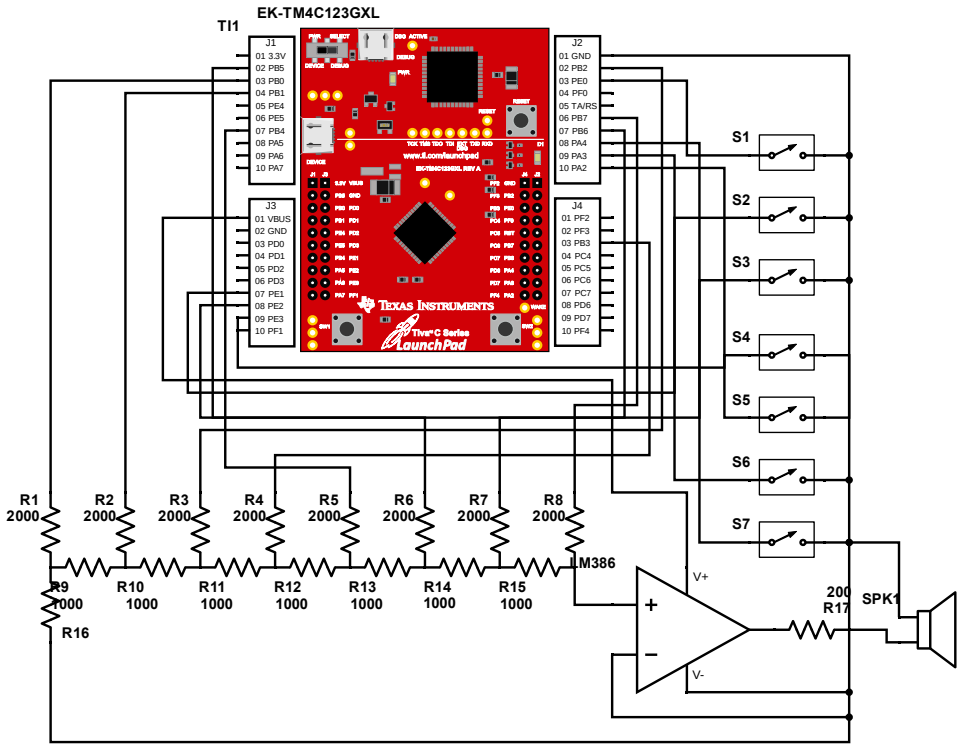
This project implements an R2R circuit that converts digital signals to analog signals. With the use of the TM4C123 controller, 8 pins of Port B are used to send digital signals to the R2R circuit. The output of the R2R signal can be sampled at a rate of the target frequency of 440Hz. We use this circuit to produce a square wave, saw tooth wave, triangle wave, and sine wave. We also use the circuit to produce 7 different sounds used as piano keys.

Operation

There are 5 modes in the project. Port F Switch 1 is used as an interrupt button to switch between modes. The following waves in modes 1-4 are produced at 440hz. In mode 1, I produced a saw tooth wave by incrementing through a variable 256 times, and at the end setting the output of the pins to 0. In mode 2, I produced a triangle wave by incrementing through 256 and then decrementing to 0. In mode 3, a square wave is produced by out putting 1s in all digital pins for some time, and then outputting 0s in all pins. A sine wave was made by incrementing through 256 values 1-256. This was produced by finding the output of each binary combination through the pins, and mapping them to their position relative to 1-256. The fifth mode was a piano key mode where the sine wave was played at different frequencies that would match the notes of a piano.

Hardware

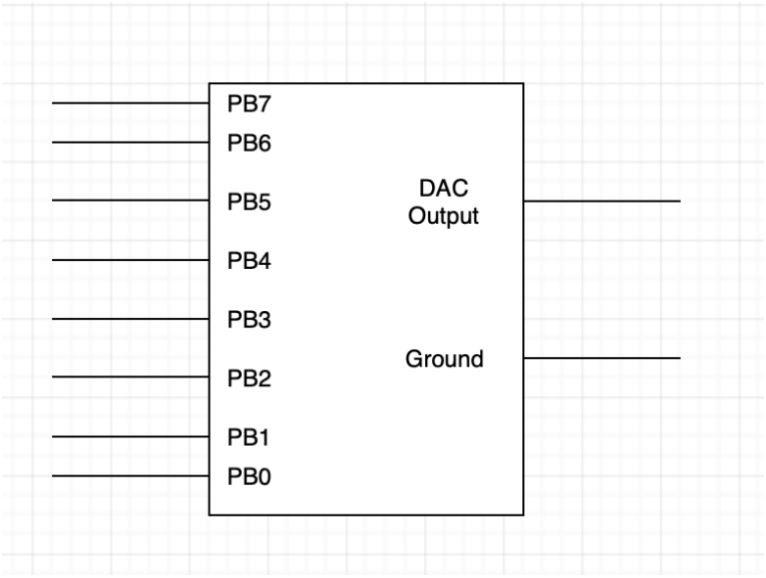
Schematics



Hardware Schematic Figure 1

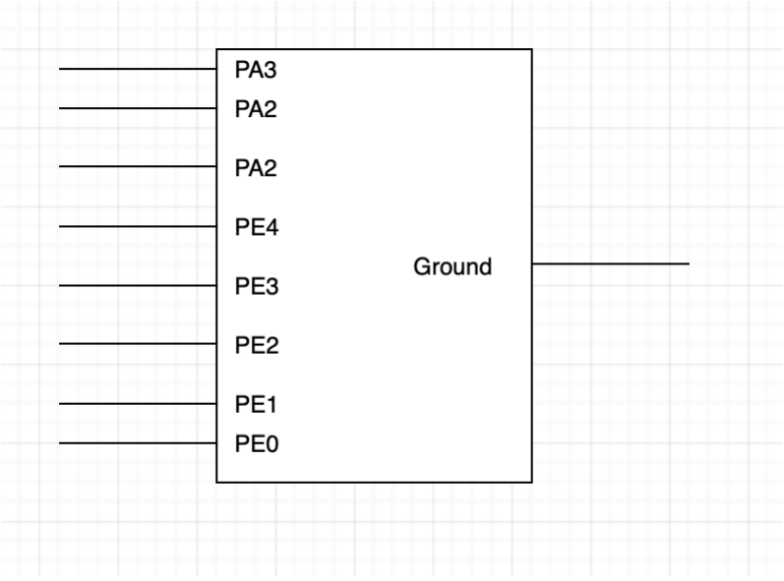
Hardware Diagram Blocks

R2R Circuit



R2R Circuit Figure 2

Button Diagram



Button Diagram Figure 3

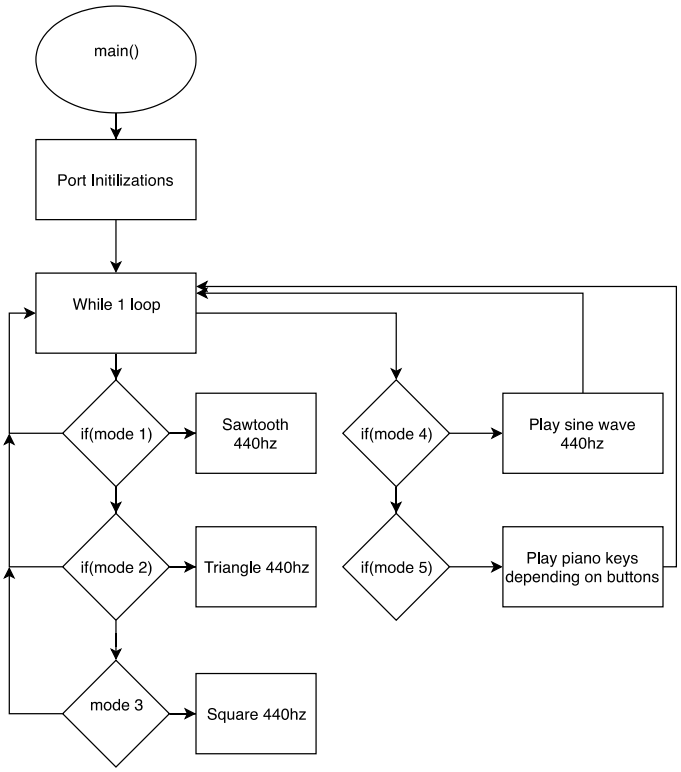
Components & Purpose

- Resistors: resistors were needed to make the R2R circuit
- Speaker: to hear the different frequencies on the piano
- Buttons: to switch between the different frequencies
- LM386: Amplifier used to amplify the current into speaker
- TM4C123: the microcontrollers used to control and send digital circuit

Software

The software approach started with trying to figure out how to produce each mode one by one. Getting the shape of the waves was not a problem, since going over the methodology in class helped my understanding in the concept. The main software challenge was initializing and using the systick timer, because I had to adjust the systick delays to the specific target frequency. After I was able to program each of the wave forms, I moved on to the piano keys. Since the sine wave was already programmed, all that was left was changing the delay and frequency according to each pitch or note we needed.

Software Flow Diagram

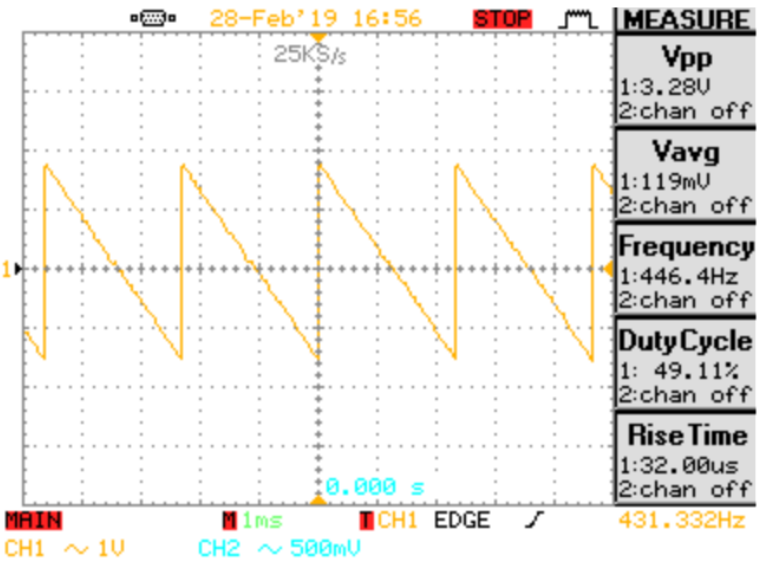


Software Flow Figure 4

Conclusion

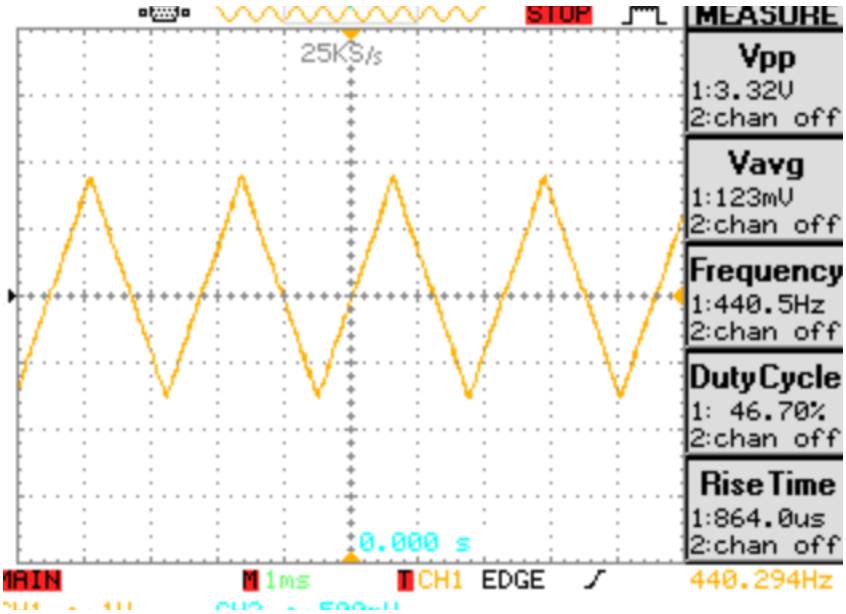
Overall, this project helped me understand and review basic microcontroller programming methods, while at the same time teaching me digital to analog conversion. Working without a partner really helped me learn more and forced me to know every single detail about the entire project. Usually working with a partner will allow me to rely on them for a certain part, but this time I was accountable for the entire project. This project pushed me in a way that I would not experience unless I worked by myself. The amount of hours put in to designing the software and creating the hardware was well worth after receiving full credit on the project. In retrospect, I've become more familiar with Systick timer/interrupts, port initializations, button interrupts, and digital to analog conversion.

Saw tooth Wave



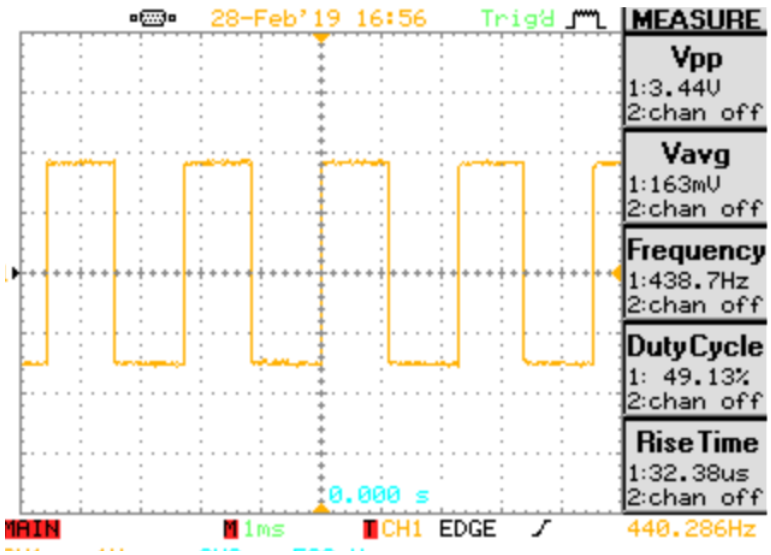
Saw Tooth Wave Figure 5

Triangle Wave



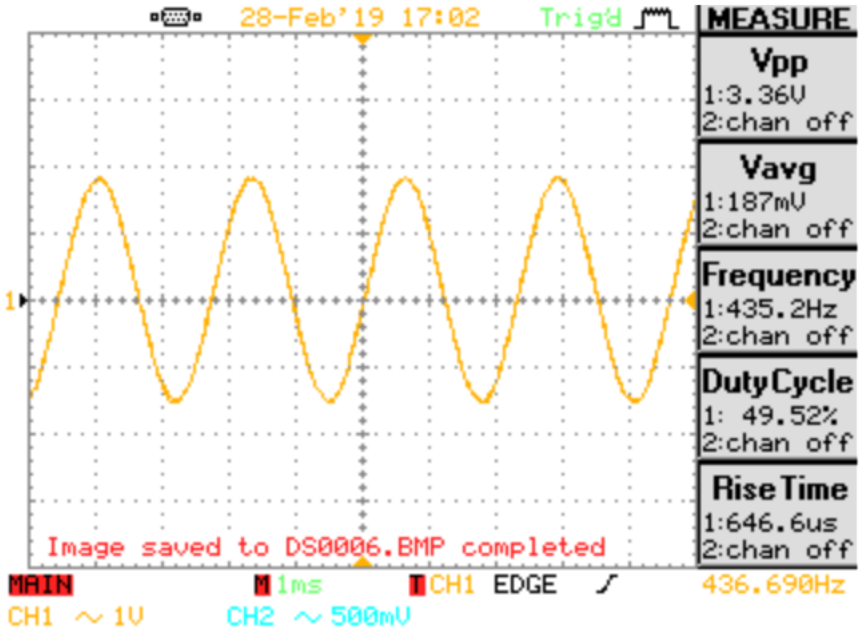
Triangle Wave Figure 6

Square Wave



Square Wave Figure 7

Sine Wave



Sine Wave Figure 8